CVS Version Management in HDL Designer Series

October 10, 2005

Introduction

With the re-write of the Concurrent Versions System (CVS) interface in HDL Designer Series, there has been a change from the previous use model. This application note walks you through setting up Version Management (VM) in HDL Designer Series, targeting CVS as your VM interface, and using the new CVS interface.

Setting up Version Management

When you want to use VM in HDL Designer Series, the first thing you need to do is make sure you have VM enabled. You do this by selecting Options -> Version Management from the menu in the Design Browser.

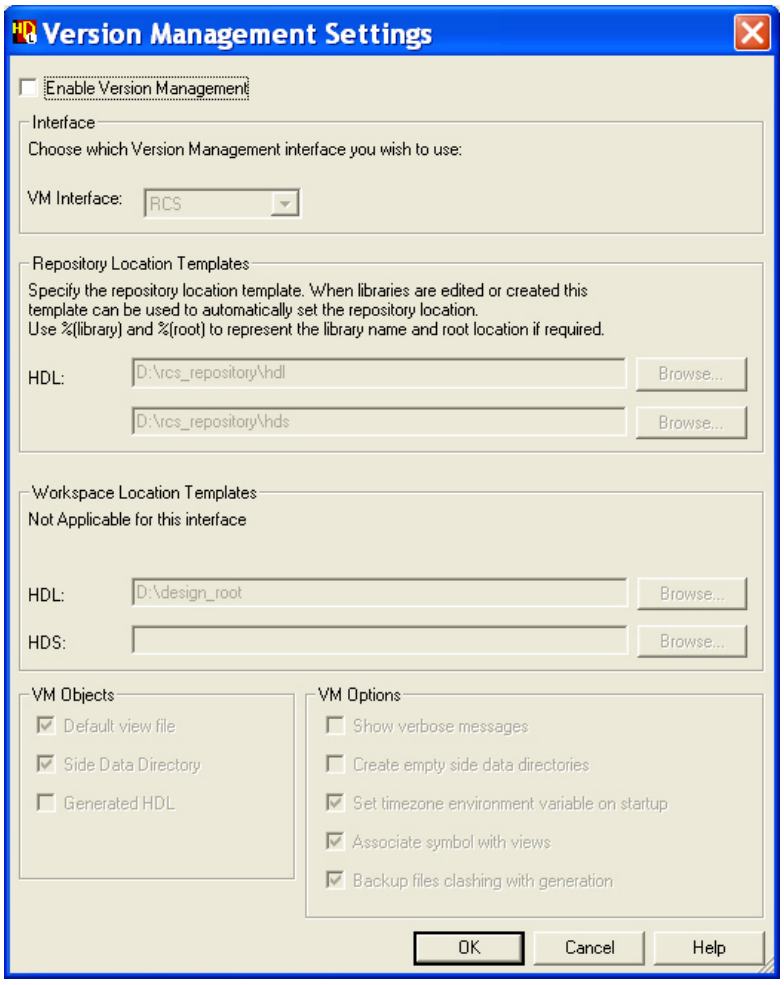


Figure 1. Version Management dialog box

 © Mentor Graphics, 1996-2005

Upon checking the box, you will have to specify from the pull down menu the VM Interface you want to use. You should select CVS. Upon selecting CVS, the look of the dialog box will change slightly.

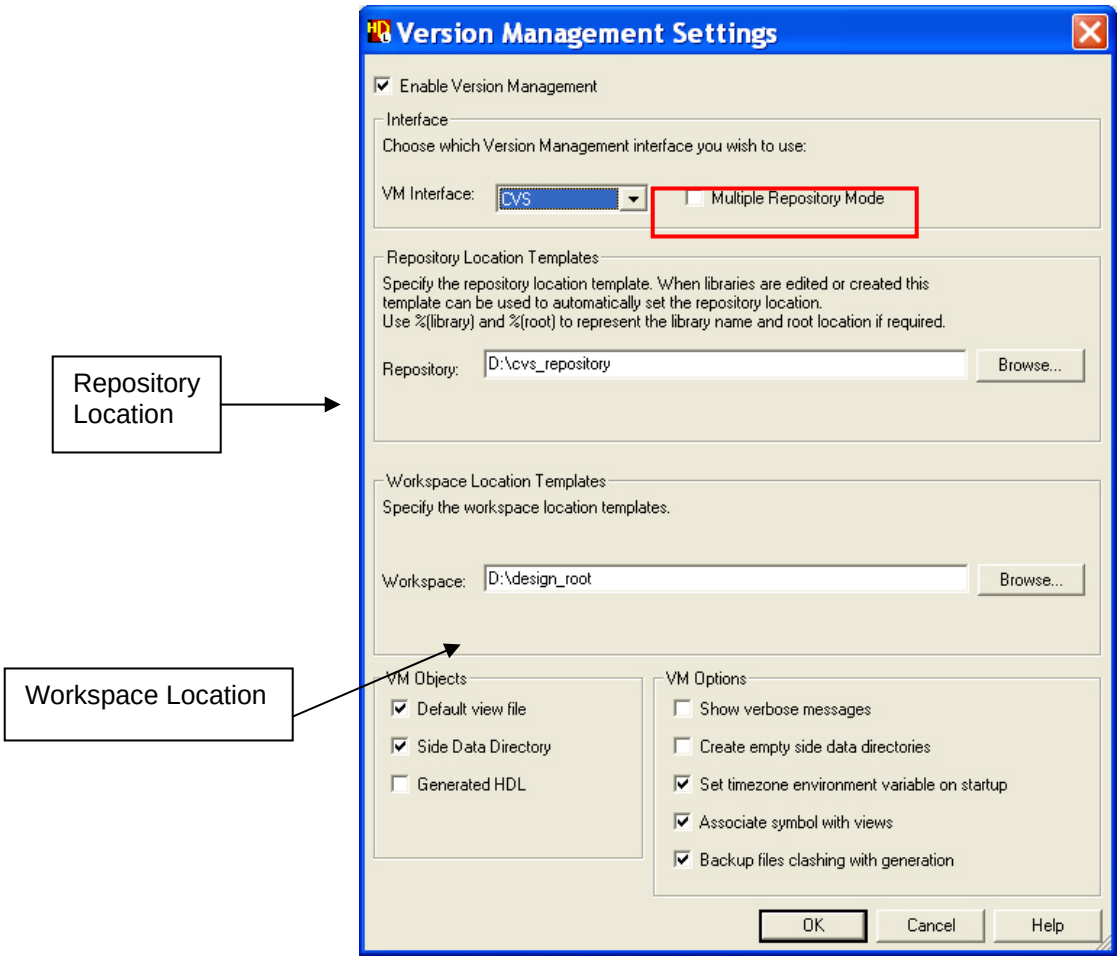


Figure 2. Version Management with CVS selected

There are two locations that have to be set up, the repository and the workspace. First, let's talk about the repository. The repository is where all your CVS VM files will be stored. It can be set up in any location, even on a remote server.

The default CVS Repository mode indicates that all the files related to your design, both text and graphical, will be stored in one repository. If you want your HDL files stored in a separate repository from your graphical-based files, you will have to check the Multiple Repository box. However, we recommend you use a single repository unless you have a specific need to have separate locations.

An improvement over the previous implementation of CVS in HDL Designer Series is how we deal with existing CVS repositories you may have. Previously, we could not use an existing repository, and would have to create a new one from scratch for your designs. Now, we can use existing repositories, and it is just a matter of pointing to it from within the dialog box. For example, if your existing CVS repository was located at C:\data\CVS_Repository, in HDL Designer Series today, you would simply specify your repository location as being C:\data\CVS_Repository. HDL Designer Series will create the necessary directory and file structure in your repository.

 © Mentor Graphics, 1996-2005

HDL Designer Series supports the additional methods of specifying a remote repository common to CVS. The first client/server mode is the pserver mode. The repository is specified in the following fashion:

[:method:] [[user]@] *hostname* [:[port]]/path/to/repository

Example: :pserver:john_smith@my_server:\projects\repository

The next type is the remote shell or rsh. For this method to work, the remote user host needs to have a '.rhosts' file which grants access to the local user. The syntax for setting up the server is:

:ext:username@servername:/cvsroot

Example: :ext:john_smith@my_server:\projects\cvs

Finally, there is the secure shell or ssh method of defining a remote repository. The syntax is the same as the rsh, except that the environment variable CVS_RSH must be set to point to the secure shell that they want to run.

Example: setenv CVS_RSH /usr/bin/ssh

Next, let's talk about the workspace. A user's CVS workspace is an area on disk where they have checked out files from a CVS repository and are modifying them, or creating new ones ready for check-in. The purpose of the workspace is twofold. First, compared to the way CVS used to work in HDL Designer Series, you now don't need to append the library name to the end of your library mapping. Second, you can now have libraries of the same name in a repository, since the location path in the repository mimics that of the workspace. What does this mean? Let us look at figure 3. If you have a library called my_lib, then the path for the HDL files will be D:\design_root\my_lib\hdl. If you looked in the repository area, the version management file will also be in a directory called my_lib\hdl.

After enabling version management for CVS, HDL Designer Series will take your workspace information into consideration when creating new libraries. The library name will be one for you to choose, but the workspace location will be pre-set as your root directory for designs. In Figure 3 below, the workspace location is specified already. As soon as you type the library name in, the source directories locations will be set as well.

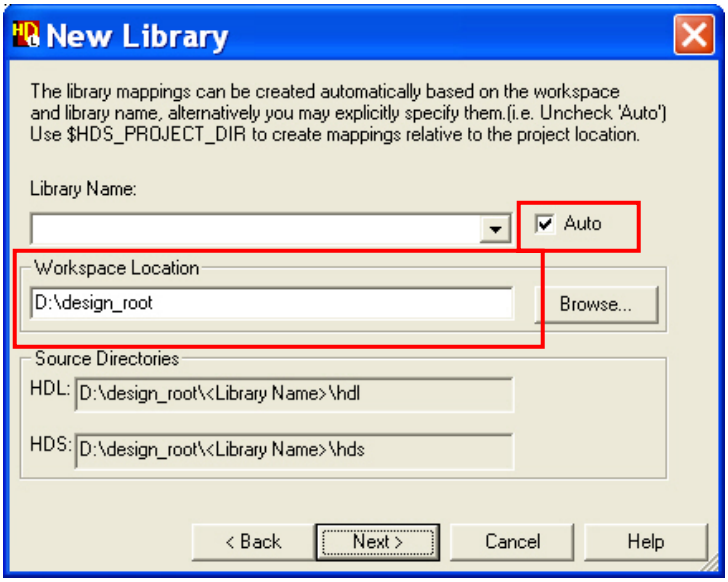


Figure 3. New library setup showing workspace

 © Mentor Graphics, 1996-2005

After giving your library a name, you will need to uncheck the “Auto” box. This is needed to confirm the version management setup for this library. After clicking “Next” twice, you will be asked whether you want to use version management.

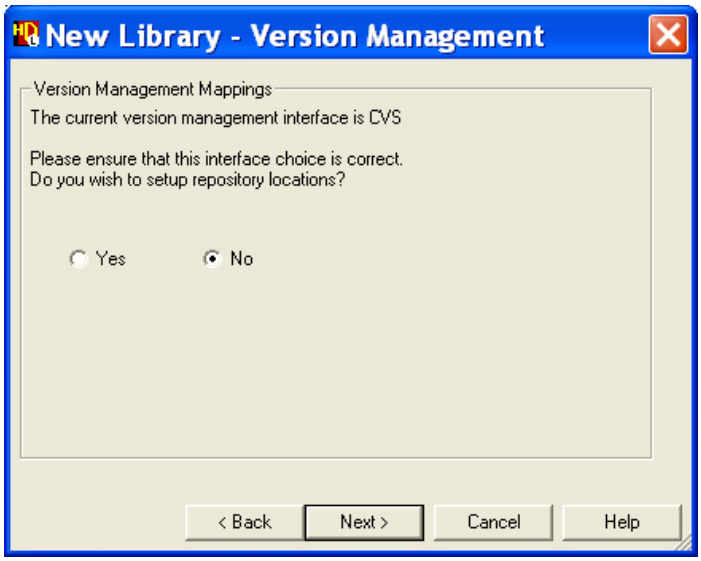


Figure 4. Setting up version management for a new library

You should go ahead and select “Yes”. The dialog box changes to indicate the repository location specified earlier.

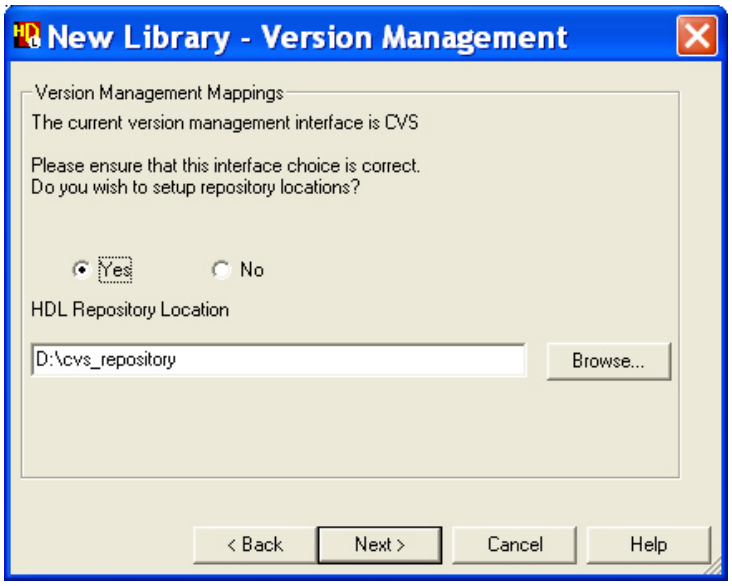
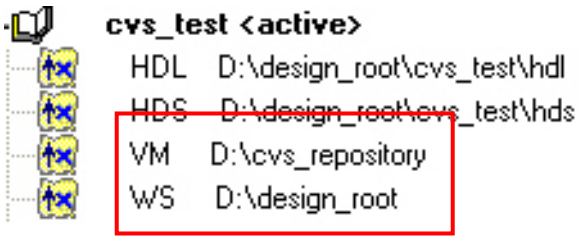


Figure 5. Library setup showing Repository location

After this you will click “next” again and finally “finish”. This completes the library setup for use with CVS version management. You will notice a change in the project manager view for that library (shown below), indicating the enabling of Version Management.

 © Mentor Graphics, 1996-2005



Using CVS in HDL Designer Series

When you have enabled CVS for VM in HDL Designer Series, you will notice the toolbar with all its functions.

Here is a summary of what all the buttons do:

Check in – The process of checking your design into the repository. This is the initial step in carrying out VM in your design. You have the option of checking in a single level, or hierarchically through the design. After check in is complete, all the design unit icons will change to indicate the design units are in a read-only mode.

Check out – When the design is in the repository, it is in a read-only mode (unless you have retained the lock when you checked it in). You need to do a check out of the design in order to make it editable in your workspace. For example, select the specific design unit and click on the check out. The icon in the design browser will change showing that it is now writable.

Get – The Get Command lets you get a copy of the latest version of the design in the repository. You can optionally choose to overwrite any design you have checked out already, and get a read only version of the entire design. This can lead to problems, so exercise caution when using the Get function.

Undo Check out – As it suggests, this undoes the previous check out.

Label – A label is a tag which you can give to specific versions in a design which may not necessarily be of the same version number. An example is if you want to create a snapshot of the design for simulation, and part of the design is at version 2.1 and another part is at version 3.1, you can check these pieces of the design in under a label “simulation”, so when anyone does a hierarchical checkout of that label, it will grab the necessary versions for simulation.

Synchronize – Synchronizes your workspace with the latest of a specified version of the design in the repository. It will only synchronize with designs which are not already checked out of the repository.

Status – Tells you the status of the design, whether it is writable, who has it locked and whether there are any labels attached

History – Gives you a history of the design through the check in/out stages. Included are comments which may have been attached to checked in versions.

 © Mentor Graphics, 1996-2005

Compare – Gives you the option to compare two versions of the code to see the differences. Only works for text design units.

Branch – The branch function is used if you need to move back to a previous version of the design, and branch off the mainline of the design. Note however, that you cannot merge the differences between the branch and the mainline. An example would be if you were on version 1.3 and you needed to go back to 1.2 and branch off it, since there was a major mistake made in 1.3, you would branch into version 1.2.1.1 and move forward from there. You also use the branch command if you just want to check out a specific version. It is a three-step process:

First, you need to synchronize the specific version you want to check out. For example, if you v1.1, 1.2 and 1.3 and you want to check out v1.2, you will click on sync, and specify v1.2
Second, you click on the “branch” button and that will start the branching process. You should give the branch a name. At this point, your design is still in read-only mode
Third, you check out the design. Now you will have v1.2. The next time you check it in, it will be the branched version. To get the latest version back, simply synchronize with the latest and do a check out.

Delete – This function is used to delete design files from within the repository directly. It is a safer way to do a delete, rather than using an Operating System method. The delete function puts the file in an “attic”, so that if you need to get it back, you can do so. However, this has to be done manually outside HDL Designer Series.

The Attic

When you delete a design unit using the VM Delete command in HDL Designer Series, CVS creates a new version of the file, and gives it the status "dead". CVS then places all versions of the design unit in a directory in the repository called the "Attic". This procedure describes how to recover a design unit from the Attic.

1. Working in your file system (outside of HDL Designer Series) sets the CVSROOT environment variable. (This procedure requires that you execute the cvs log command and the cvs update command, either in a DOS shell or a UNIX/Linux shell. The CVSROOT environment variable setting provides the repository location for these cvs commands.) Set CVSROOT to the same directory specification that appears for the VM specification in your library mapping. For example: C:/apps/hds2004_1a/examples/hds_scratch/hds_repository/hdl_vm.
To observe the VM directory specification, go to the Project tab in HDL Designer Series and expand the library mappings for the library you are working with. Then set CVSROOT as follows:

On Windows:

```
set CVSROOT=C:/apps/hds2004_1a/examples/hds_scratch/hds_repository/hdl_vm
```

On Unix or Linux

```
setenv CVSROOT C:/apps/hds2004_1a/examples/hds_scratch/hds_repository/hdl_vm
```

2. Note that this procedure cannot be carried out in HDL Designer Series, but instead has to be done by command line. At a DOS prompt, go into the workspace directory where the file was deleted from. (This will be the directory in your **HDS workspace** (not the repository) where the design unit existed before it was deleted. For example, if the design unit was purely textual, then go into the *hdl* directory for the library. If the design unit was graphical, then go into the *design unit subdirectory* of the hds directory. For example, if the design unit was a state diagram named "controller", then you would go into the hds\controller directory.

3. Execute the following command:

```
cvs log <filename>
```

 © Mentor Graphics, 1996-2005

For example, if the design unit was a textual file called status_registers_spec.vhd, then you would type in the following:

```
cvs log status_registers_spec.vhd
```

If the design unit was a state machine called "controller", then you would type in:

```
cvs log fsm.sm
```

This command gives you a status display such as the following, showing all versions of the file:

```
RCS file:C:/<path to Repository>/hdl_vm/UART/hdl/Attic/status_registers_spec.vhd,v
Working file: status_registers_spec.vhd
head: 1.4
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
total revisions: 4;        selected revisions: 4
description:
-- no description --
----------------------------
revision 1.4
date: 2004/11/17 20:52:04;  author: rhansen;  state: dead;  lines: +0 -0
"Removed by HDS"
----------------------------
revision 1.3
date: 2004/11/17 20:51:56;  author: rhansen;  state: Exp;  lines: +1 -0
-- no description --
----------------------------
revision 1.2
date: 2004/11/17 20:41:38;  author: rhansen;  state: Exp;  lines: +0 -0
-- no description --
----------------------------
revision 1.1
date: 2004/09/22 15:41:20;  author: rhansen;  state: Exp;
-- no description --
=============================================================================
```

The last version will have the status "dead". Pick the "non-dead" version that you want to recover. For example, using the above display, you might decide to pick revision 1.3.

NOTE: If you are unsure of the exact file name to specify in the cvs log command you can determine the file name by looking in the appropriate Attic directory in the repository. For example, for a graphical design unit named “controller”, you would look in the following location:

```
<path-to-repository>\<library-name>\hds\controller\Attic
```

In this directory you will see the repository files for this design unit. For example:

```
fsm.sm,c
symbol.sb,v
```

 © Mentor Graphics, 1996-2005

The exact file names to specify in the cvs commands are the file names that you see here, minus the ,v extensions.

4. After picking the version that you want to recover, and using the above example, type in the following command:

cvs update -p -r1.3 status_registers_spec.vhd > status_registers_spec.vhd

Note that you use the -r argument to specify the version you want to recover.

CVS gives a message such as the following:

```
Checking out status_registers_spec.vhd
RCS:  <path-to-repository>/UART/hdl/Attic/status_registers_spec.vhd,v
VERS: 1.3
******************
```

For a graphical design unit, you would type in the command:

cvs update -p -r1.3 fsm.sm > fsm.sm

and

cvs update -p -r1.3 symbol.sb > symbol.sb

separately to recover a state machine.

5. Execute View>Refresh in the Design Explorer in HDL Designer Series.

At this point, you should see that the design unit is back in the library, and you can now edit it and check it in as normal.

Note that you need to execute the cvs update command once for each file that you want to recover. For example, for a state machine component, you would execute the cvs update command once for the symbol.sb file and once again for the fsm.sm file.

Conclusion

You should now be on your way to using CVS for version management in HDL Designer Series. You can find additional information about version management with CVS in the help section of HDL Designer Series, under the User Manual subsection called Version Management. You can also find a CVS manual in PDF form in the “help” directory of your HDL Designer Series software installation.

Other Information

For additional information about HDL Designer Series, please visit http://www.hdldesigner.com, or send email to hdldesigner info@mentor.com

 © Mentor Graphics, 1996-2005